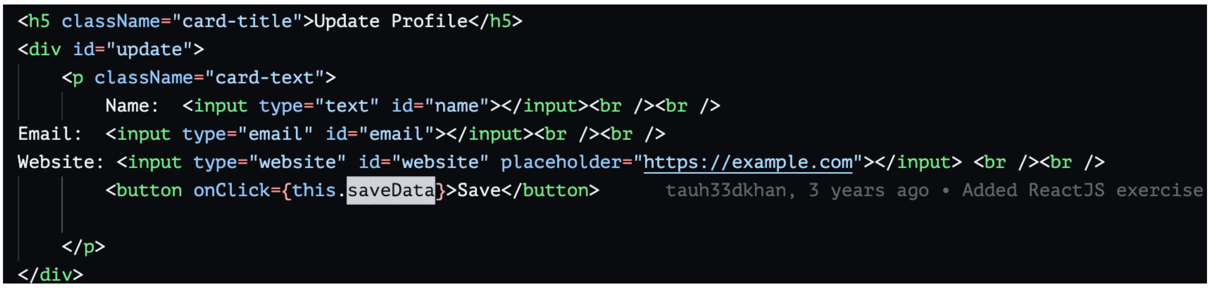
Path

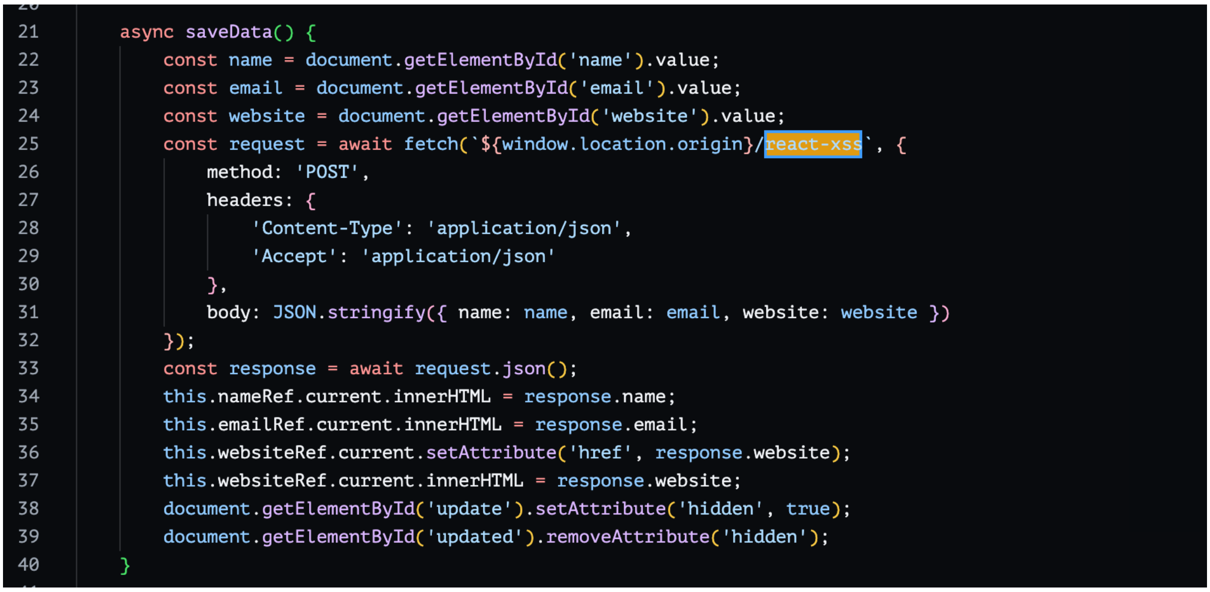
vuln_react_app/src/MyComponents/React_ref_innerHTML_xss.js

saveData() gets called whenever the save button gets clicked.



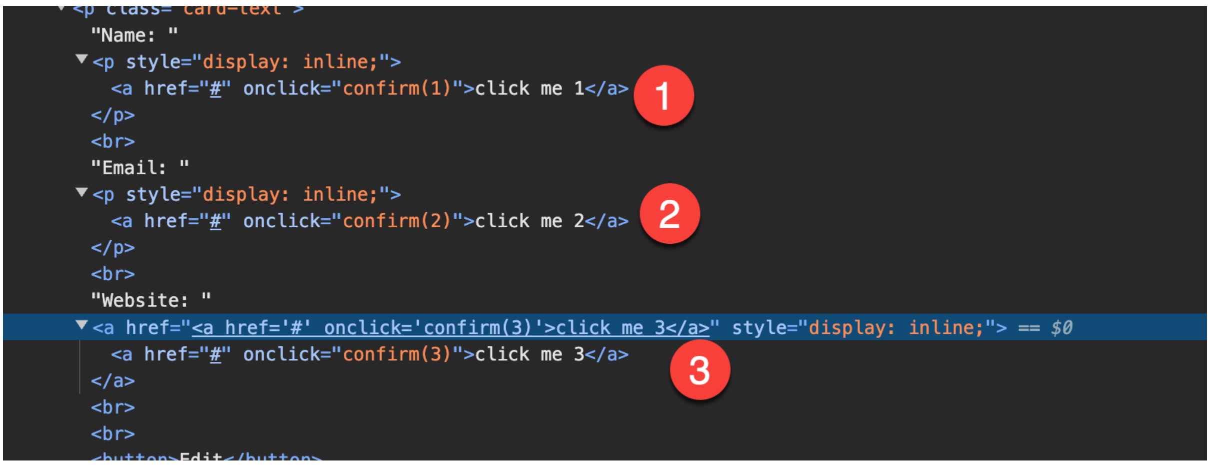
```
<h5 className="card-title">Update Profile</h5>
<div id="update">
    <p className="card-text">
        Name:  <input type="text" id="name"></input><br /><br />
Email:  <input type="email" id="email"></input><br /><br />
Website: <input type="website" id="website" placeholder="https://example.com"></input> <br /><br />
        <button onClick={this.saveData}>Save</button>      tauh33dkhan, 3 years ago • Added ReactJS exercise

    </p>
</div>
```

The issue right here is that innerHTML gets used and as such it is possible to get insert XSS payload on name, email and website.

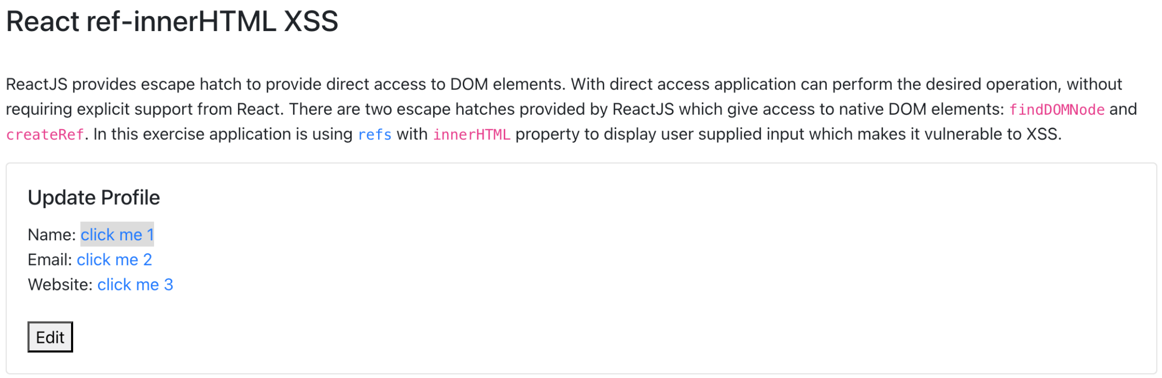


```
    async saveData() {
        const name = document.getElementById('name').value;
        const email = document.getElementById('email').value;
        const website = document.getElementById('website').value;
        const request = await fetch(`${window.location.origin}/react-xss`, {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
                'Accept': 'application/json'
            },
            body: JSON.stringify({ name: name, email: email, website: website })
        });
        const response = await request.json();
        this.nameRef.current.innerHTML = response.name;
        this.emailRef.current.innerHTML = response.email;
        this.websiteRef.current.setAttribute('href', response.website);
        this.websiteRef.current.innerHTML = response.website;
        document.getElementById('update').setAttribute('hidden', true);
        document.getElementById('updated').removeAttribute('hidden');
    }
```

In this case, a harmless XSS payload was used so that whenever a link gets clicked, a confirm button will pop.



```
  ▼<p class="card-text">
      "Name: "
     ▼<p style="display: inline;">
        <a href="#" onclick="confirm(1)">click me 1</a>
      </p>
      <br>
      "Email: "
     ▼<p style="display: inline;">
        <a href="#" onclick="confirm(2)">click me 2</a>
      </p>
      <br>
      "Website: "
     ▼<a href="<a href='#' onclick='confirm(3)'>click me 3</a>" style="display: inline;"> == $0
        <a href="#" onclick="confirm(3)">click me 3</a>
      </a>
      <br>
      <br>
      <button>Edit</button>
```

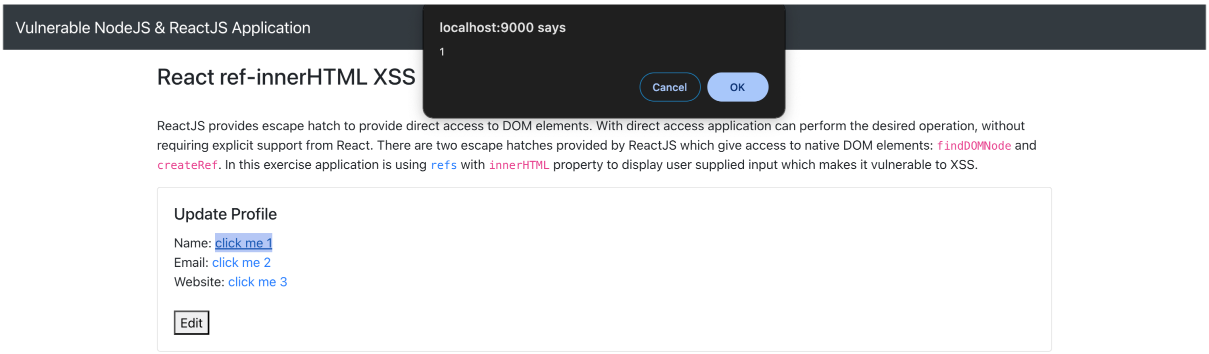


React ref-innerHTML XSS

ReactJS provides escape hatch to provide direct access to DOM elements. With direct access application can perform the desired operation, without requiring explicit support from React. There are two escape hatches provided by ReactJS which give access to native DOM elements: `findDOMNode` and `createRef`. In this exercise application is using `refs` with `innerHTML` property to display user supplied input which makes it vulnerable to XSS.

Update Profile

Name: click me 1
Email: click me 2
Website: click me 3



localhost:9000 says

1

React ref-innerHTML XSS

ReactJS provides escape hatch to provide direct access to DOM elements. With direct access application can perform the desired operation, without requiring explicit support from React. There are two escape hatches provided by ReactJS which give access to native DOM elements: `findDOMNode` and `createRef`. In this exercise application is using `refs` with `innerHTML` property to display user supplied input which makes it vulnerable to XSS.

Update Profile

Name: click me 1
Email: click me 2
Website: click me 3